

Amendments to the Claims

This listing of claims will replace all prior versions (and listings) of claims without prejudice in the application.

Listing of Claims

1. (Currently amended) A computer program embodied on a computer readable medium for creating building a knowledge-oriented software application using an object-oriented programming language, the computer program comprising:

a ~~Meta Knowledge Model~~ code segment ~~comprising for storing a means for representing knowledge in a human readable format a set of logic details~~ for the knowledge-oriented software application;

~~a Meta Logic Model comprising a means for deriving new knowledge in the knowledge-oriented software application~~ a code segment for storing a set of relation functions for deriving new relations from a set of base relations and imposing rules and constraints on the relations, wherein the new relations comprise the relation functions providing code that accepts a predetermined number of base relations as input and returns derived knowledge as data represented as a relation; including code for the relation function to accept a predetermined number and type of inputs and calculate a derived knowledge data;

a code segment for Knowledge Definition Model comprising storing a definition and a set of requirements for each of the relation functions in human readable documentation of the relation function that is used to create the knowledge-oriented software application, wherein the Knowledge Definition Model receives a definition and a predetermined number of set requirements, and further wherein the requirements include type information, instance

~~information and binding information, a means for converting the knowledge-oriented software application from a human-readable format to a computer-executable process;~~

~~a Catalog Model code segment for storing within an electronic device, comprising a means for modeling a set of user sessions for the knowledge-oriented software application, a logical design of the knowledge-oriented software application, and run time data for the knowledge-oriented software application; and~~

~~a Run-time code segment Model comprising a means for dynamically assembling a set of executable components at run time, wherein the set of executable components converts the set of logic details from human readable format into code that is executable by a computer;~~

~~enabling concurrent clients to execute the knowledge-oriented software application to manipulate the knowledge stored in the electronic device, wherein the Run-time Model executes the relation function based on the definition and requirements inputted in the Knowledge Definition Model.~~

2. (Currently amended) The computer program of claim 1, wherein the Meta-Knowledge Model ~~code segment for storing in a human readable format a set of logic details~~ further comprises a Thing, a Key, a Fact, ~~a Relation~~, and a Knowledge Definition.

3. (Previously presented) The computer program of claim 2, wherein the Thing comprises an object-oriented representation of a named object.

4. (Previously presented) The computer program of claim 2, wherein the Key comprises an object-oriented representation of a name of the Thing.

5. (Previously presented) The computer program of claim 2, wherein the Fact comprises an object-oriented representation of facts regarding a set of things.

6. (Currently amended) The computer program of claim ~~21~~, wherein the ~~R~~relation functions further comprises ~~an~~ object-oriented representations of a set of facts.

7. (Canceled)

8. (Currently amended) The computer program of claim 1, wherein the ~~Meta-Logic-Model~~ code segment for storing a set of relation functions further comprises an Event, and an Event Handler, ~~and a Relation Function~~.

9. (Previously presented) The computer program of claim 8, wherein the Event comprises an object-oriented representation of functions selected from a group consisting of: adding, editing, modifying, and dropping a Relation.

10. (Previously presented) The computer program of claim 8, wherein the Event Handler comprises an object oriented representation of responses to the Event.

11. (Currently amended) The computer program of claim ~~81~~, wherein the set of Rrelation ~~F~~functions comprises an object-oriented representations of actions.

12. (Currently amended) The computer program of claim ~~81~~, wherein the set of Rrelation ~~F~~functions comprises an object-oriented representations of queries.

13. (Currently amended) The computer program of claim ~~81~~, wherein the set of Rrelation ~~F~~functions comprises an object-oriented representations of functions.

14 - 29 (Canceled)

30. (Currently amended) A computerized system for ~~building~~creating a knowledge-oriented software application using an object-oriented programming language comprising:

a processor; and

a storage device comprising:

~~a Meta-Knowledge Model comprising a means for representing knowledge~~code segment for storing in a human readable format a set of logic details for the knowledge-oriented software application, and further comprising a Thing, a Key, a Fact, a Relation, and a Knowledge Definition, wherein the Thing comprises an abstract base Thing class with a method for retrieving a kKey of a Thing instance, further wherein the kKey comprises a unique identifier;

~~a Meta-Logic Model comprising a means for deriving new knowledge in the knowledge-oriented software application~~code segment for storing a set of relation functions for deriving new relations from a set of base relations and imposing rules and constraints on the relations, wherein the new relations comprise the relation functions providing code that accepts a predetermined number of base relations as input and returns derived knowledge as data represented as a relation;

~~a Knowledge-Definition Model comprising a means for converting the knowledge-oriented software application from a human-readable format to a computer-executable process~~code segment for storing a definition and a set of requirements for each of the relation functions in the knowledge-oriented software application;

~~a Catalog Model comprising a means for modeling~~code segment for storing a set of user sessions for the knowledge-oriented software application, a logical design of the knowledge-oriented software application, and run time data for the knowledge-oriented software application within an electronic device; and

a Run-time Model comprising a means code segment for dynamically assembling a set of executable components at run time, wherein the set of executable components converts the set of logic details from human readable format into code that is executable by a computer;

enabling concurrent clients to execute the knowledge-oriented software application to manipulate the knowledge stored in the storage device.

31. (Previously presented) The computerized system of claim 30, wherein the Thing comprises a static Thing Definition object that defines its physical implementation.

32. (Previously presented) The computerized system of claim 31, wherein the Thing is extendible to create application-specific Thing classes and wherein the Thing class is a base class for all domain-specific object classes.

33. (Previously presented) The computerized system of claim 30, wherein the Key comprises an abstract base type and a StringKey, a DateKey and a IntegerKey are sub types of the Key.

34. (Previously presented) The computerized system of claim 33, wherein the StringKey creates the Key from a string and wherein the StringKey comprises a method to get a string value of a string key.

35. (Previously presented) The computerized system of claim 33, wherein the IntegerKey creates the Key from an integer and wherein the IntegerKey comprises a method to get an integer value of an integer key.

36. (Previously presented) The computerized system of claim 33, wherein the DateKey creates the Key from a date and wherein the DateKey comprises a method to get a date value of a date key, and wherein the Key is extendible to create other types of keys.

37. (Previously presented) The computerized system of claim 30, where the Fact is a sub type of the Thing.

38. (Previously presented) The computerized system of claim 30, wherein the Fact is as an abstract base type.

39. (Previously presented) The computerized system of claim 30, wherein the Fact is a set of the Things, and wherein the Things can be gotten and updated by their ordinal numbers.

40. (Previously presented) The computerized system of claim 30, wherein the Fact is a set of the Things, and wherein the Things can be gotten and updated by their names.

41. (Previously presented) The computerized system of claim 30, wherein an implementation of the Fact has a fixed cardinality.

42. (Previously presented) The computerized system of claim 30, wherein the Fact has a Type and methods to get and set the Fact Type, wherein the Fact Type can be Simple, OneToOne, and ManyToOne.

43. (Previously presented) The computerized system of claim 39, wherein the Things of the Fact can be gotten and set as an array of the Things.

44. (Withdrawn) A signal-bearing medium comprising a Meta Model for building a knowledge-oriented software application using an object-oriented programming language, wherein the Meta Model comprises:

a Meta Knowledge Model including:

a Thing comprising a specification of how the knowledge-oriented software application is represented as objects, wherein the Thing is the super class of the objects;

a Key comprising a specification of how knowledge-oriented software application things are named, where the thing names are instances of classes that are subclasses of the Key and the things are instances of subclasses of the Thing;

a Fact comprising a specification of how knowledge oriented software application represent facts, where the facts are instances of classes that are subclasses of the Fact and the Fact is a subclass of the Thing,

a Relation comprising a specification of how the knowledge-oriented software application represents sets of similar facts, where all relations are instances of classes that are subclasses of the Relation, and the Relation is a subclass of Thing, and

a Knowledge Definition comprising a specification of how the knowledge-oriented software application represents the logical design of the application and how it relates the logical design to a physical implementation, wherein the Knowledge Definition is a subclass of the Thing.

45. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation is a super type of the Thing.

46. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation is as an abstract base type.

47. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation is a set of named Facts.

48. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation has a Relation Scope.

49. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation has methods to get and set the Relation Scope.

50. (Withdrawn) The signal-bearing medium of claim 48, wherein the Relation Scope comprises an Application and a Session.

51. (Withdrawn) The signal-bearing medium of claim 44, where each instance of the Relation comprises a name, and wherein the Relation has methods to get and set the name.

52. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation comprises methods to retrieve, insert, delete, and update facts of the Relation.

53. (Withdrawn) The signal-bearing medium of claim 44, wherein Relation has methods to find the union, intersections, and the difference of two relations.

54. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation comprises a method to determine whether a fact is contained within a relation.

55. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation comprises a method to get the number of facts in a relation.

56. (Withdrawn) The signal-bearing medium of claim 44, wherein the Relation comprises a method to return all the facts as an array.

57. (Withdrawn) The signal-bearing medium of claim 44, wherein the Knowledge Definition is a sub type of the Thing.

58. (Withdrawn) The signal-bearing medium of claim 44, wherein the Knowledge Definition is an abstract base type.

59. (Withdrawn) The signal-bearing medium of claim 44, wherein the Knowledge Definition has a name, and wherein the Knowledge Definition comprises a method to get and set the name.

60. (Withdrawn) The signal-bearing medium of claim 44, wherein the Knowledge Definition comprises a logical and physical representation of a Thing.

61. (Withdrawn) A signal-bearing medium comprising a Meta Model for computer programs implemented in an object-oriented programming language, the Meta Model comprising:

a Meta Knowledge Model comprising a Thing, a Key, a Fact, a Relation, and a Knowledge Definition; and

a Meta Logic Model comprising an Event, an Event Handler, and a Relation Function, wherein the Event is a sub type of the Thing.

62. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event has a name, and wherein the Event has methods to get and set the name.

63. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event has a Level and wherein the Event has methods to get and set the Event Level, and wherein the Event Level can be selected from a group consisting of: the Relation and the Fact.

64. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event has a Priority, and wherein the Event has methods to get and set the Event Priority.

65. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event has a Scope, and wherein the Event has methods to get and set the Event Scope, and wherein the Event Scope can be selected from a group consisting of: an Application or a Session.

66. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event has a Type, wherein the Event has methods to get and set the Event Type, and wherein the Event Type can be selected from a group consisting of: On Insert, On Delete, On Update, On Add, and On Drop.

67. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event has methods to get and set the Event Handler.

68. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event Handler comprises a sub type of the Thing.

69. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event Handler comprises:

a name; and

methods to get and set the name.

70. (Withdrawn) The signal-bearing medium of claim 61, wherein the

Event Handler comprises:

a relation function; and

methods to get and set a name of the relation function.

71. (Withdrawn) The signal-bearing medium of claim 61, wherein the

Event Handler comprises:

a target relation; and

methods to get and set a name of the target relation.

72. (Withdrawn) The signal-bearing medium of claim 61, wherein the Event Handler comprises:

an action type; and

methods to get and set the action type, wherein the action type is selected from a group consisting of: Clear, Add, Drop, Replace, Insert, Delete, and Update.

73. (Withdrawn) The signal-bearing medium of claim 61, wherein the Relation Function is a super type of the Thing.

74. (Withdrawn) The signal-bearing medium of claim 61, wherein the Relation Function comprises:

a name; and

methods to get and set the name.

75. (Withdrawn) The signal-bearing medium of claim 61, wherein the Relation Function has an argument and the argument is an array of the Relations.

76. (Withdrawn) The signal-bearing medium of claim 61, wherein the Relation Function has a return value and the return value is a Relation.

77. (Withdrawn) The signal-bearing medium of claim 61, wherein the Relation Function defines a method to execute.

78. (Withdrawn) The signal-bearing medium of claim 61, wherein the Relation Function and the Relation form a closed algebra, wherein the Relation Function returns only a relation,

and wherein the Relation Function can be used as an argument anywhere a Relation can be used as an argument.

79. (Withdrawn) The signal-bearing medium of claim 61, wherein the Relation Function is recursive, wherein an output of the Relation Function is an input of the Relation Function, and wherein the Relation Function terminates when an output Relation is empty.

80. (Withdrawn) A signal-bearing medium comprising:

a Meta Model for computer programs implemented in an object-oriented programming language, the Meta Model comprising:

a Meta Knowledge Model comprising a Thing, a Key, a Fact, a Relation, and a Knowledge Definition, and

a Meta Logic Model comprising an Event, an Event Handler, and a Relation Function; and

a Kernel, wherein the Kernel is a super type of the Thing.

81. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel comprises a system Application that manages external applications.

82. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel comprises a name and methods to get and set the name.

83. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel comprises a root object of all objects.

84. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel has a number of users and methods to add and drop users identified by their names and passwords.

85. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel grants access to Applications and wherein the Kernel comprises methods to grant privileges to users.

86. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel has multiple concurrent users, wherein each of the users has multiple sessions, and wherein the Kernel comprises methods to create, get, and drop named sessions.

87. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel comprises methods to load and unload an Application.

88. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel loads an application from an Application Definition File.

89. (Withdrawn) The signal-bearing medium of claim 80, wherein the signal-bearing medium further comprises an Application.

90. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application is a sub type of the Thing.

91. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a version and the Application has methods to get and set the version.

92. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a name and the Application has methods to get and set the name.

93. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a set of Relation Definition instances and the Application has methods to add, get, and drop the Relation Definition instances.

94. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a set of Relation instances and the Application has methods to add, get, and drop the Relation instances.

95. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a set of Relation Function Definition instances and the Application has methods to add, get, and drop the Relation Function Definition instances.

96. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a set of Relation Function instances and the Application has methods to add, get, and drop the Relation Function instances.

97. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a set of Event instances and the Application has methods to add, get, and drop the Event instances.

98. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a set of Event Handler instances and the Application has methods to add, get, and drop the Event Handler instances.

99. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application has a set of Thing Definition instances and the Application has methods to add, get, and drop the Thing Definition instances.

100. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel enforces type safety.

101. (Withdrawn) The signal-bearing medium of claim 100, wherein when a relation, relation function, thing, key, and fact are created and referenced, the Kernel enforces the type specified in a corresponding knowledge definition.

102. (Withdrawn) The signal-bearing medium of claim 80, wherein the Kernel enforces referential integrity.

103. (Withdrawn) The signal-bearing medium of claim 102, where when a relation, relation function, thing, key, and fact are referenced by the key, the Kernel enforces that an object referenced by the key exists in the relation specified by the key's reference definition.

104. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application manages a transaction, wherein a transaction comprises a change in a state of a relation.

105. (Withdrawn) The signal-bearing medium of claim 104, wherein the Application has a method to start the transaction.

106. (Withdrawn) The signal-bearing medium of claim 104, wherein the Application has a method to commit the transaction.

107. (Withdrawn) The signal-bearing medium of claim 104, wherein the Application has a method to rollback the transaction.

108. (Withdrawn) The signal-bearing medium of claim 104, wherein the Application has a method to execute the Relation Function and return a result as a relation.

109. (Withdrawn) The signal-bearing medium of claim 104, wherein the Application rolls back an external transaction when the external transaction fails.

110. (Withdrawn) The signal-bearing medium of claim 109, wherein the Application rolls back an external transaction when a transaction of an event generated by the external transaction fails.

111. (Withdrawn) The signal-bearing medium of claim 109, wherein the Application commits the external transaction and the transactions generated by the events as a result of the external transaction.

112. (Withdrawn) The signal-bearing medium of claim 89, wherein the Application is persistently and declaratively stored in an Application Definition File.

113. (Withdrawn) The signal-bearing medium of claim 112, wherein a Kernel binds the Application Definition to a physical implementation at run time.

114. (Withdrawn) The signal-bearing medium of claim 80, wherein a Session is a sub type of the Thing.

115. (Withdrawn) The signal-bearing medium of claim 114, wherein the Session has a name and the Session has methods to get and set the name.

116. (Withdrawn) The signal-bearing medium of claim 114, wherein the Session has methods to access any Kernel application.

117. (Withdrawn) The signal-bearing medium of claim 114, wherein the Session has a time out and wherein the Kernel removes timed-out sessions automatically.

118. (Withdrawn) The signal-bearing medium of claim 114, wherein the Session has a state.

119. (Currently amended) The computer program of claim 1, wherein the ~~Catalog-Model~~ comprises a meanscode segment for storing a set of user sessions for the knowledge-oriented application further comprises ~~for~~ storing a declarative definition of an Application and the Application's run-time data.

120. (Currently amended) The computer program of claim 119, wherein the ~~Catalog-Model~~ code segment for storing a set of user sessions for the knowledge-oriented application further comprises a set of classes that are subclasses of the Thing, Key, ~~R~~relation functions, Fact, and Knowledge Definition.

121. (currently amended) The computer program of claim 1, wherein the ~~Run-time Model~~ code segment for dynamically assembling the set of executable components at run time further comprises a Session, a Kernel, and subtypes of an Application.

122. (Previously presented) The computer program of claim 121, wherein the Application comprises an object oriented definition of how the Application can be accessed by a computer program that uses the services of the Application.

123. (Previously presented) The computer program of claim 121, wherein the Session comprises an object oriented definition of how the Application can be accessed by concurrent users.

124 - 136 (Canceled)

137. (Currently amended) The computerized system of claim ~~136~~30, wherein the Thing comprises an object-oriented representation of a named object.

138. (Currently amended) The computerized system of claim ~~136~~30, wherein the Key comprises an object-oriented representation of a name of the Thing.

139. (Currently amended) The computerized system of claim ~~136~~30, wherein the Fact comprises an object-oriented representation of facts regarding a set of things.

140. (Currently amended) The computerized system of claim ~~136~~30, wherein the ~~R~~relation functions further comprises ~~an~~ object-oriented representations of a-sets of facts.

141. (Currently amended) The computerized system of claim 30, wherein the ~~Run-time Model~~code segment for dynamically assembling a set of executable components at run time further comprises a Session, a Kernel, and subtypes of an Application.

142. (Previously presented) The computerized system of claim 141, wherein the Application comprises an object oriented definition of how the Application can be accessed by a computer program that uses the services of the Application.

143. (Previously presented) The computerized system of claim 141, wherein the Session comprises an object oriented definition of how the Application can be accessed by concurrent users.

144 - 148 (Canceled)